https://p4.org/p4/getting-started-with-p4.html

=====================================================================================

sudo apt update

sudo apt-get install -y cmake g++ git automake libtool libgc-dev bison flex libfl-dev libgmp-dev libboost-dev libboost-iostreams-dev libboost-graph-dev llvm pkg-config python python-scapy python-ipaddr python-ply tcpdump doxygen graphviz texlive-full

sudo apt-get install autoconf autogen automake libtool python-pip

=====================================================================================

git clone https://github.com/protocolbuffers/protobuf.git

cd protobuf

git checkout v3.2.0

git submodule update --init --recursive

./autogen.sh

./configure

make

make check

sudo make install

sudo ldconfig

=====================================================================================

git clone https://github.com/p4lang/behavioral-model.git

cd behavioral-model

./install\_deps.sh

./autogen.sh

./configure

sudo make

sudo make install

sudo ldconfig

=====================================================================================

git clone --recursive https://github.com/p4lang/p4c.git

cd ~/p4c

mkdir build

cd build

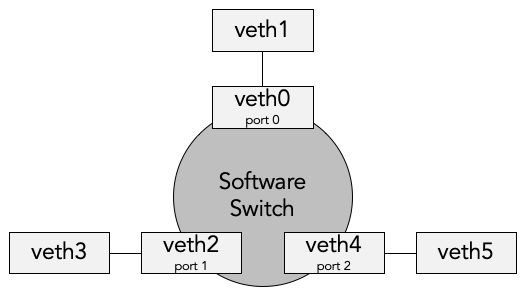
cmake ..

make -j4

make -j4 check

sudo make install

=====================================================================================



veth33 (3)

veth11 (1)

veth22 (2)

veth30 (0)

veth10 (0)

veth20 (0)

# First pair: veth10-veth11

sudo ip link add name veth10 type veth peer name veth11

sudo ip link set dev veth10 up

sudo ip link set dev veth11 up

sudo ip link set veth10 mtu 9500

sudo ip link set veth11 mtu 9500

sudo sysctl net.ipv6.conf.veth10.disable\_ipv6=1

sudo sysctl net.ipv6.conf.veth11.disable\_ipv6=1

# Second pair: veth20-veth22

sudo ip link add name veth20 type veth peer name veth22

sudo ip link set dev veth20 up

sudo ip link set dev veth22 up

sudo ip link set veth20 mtu 9500

sudo ip link set veth22 mtu 9500

sudo sysctl net.ipv6.conf.veth20.disable\_ipv6=1

sudo sysctl net.ipv6.conf.veth22.disable\_ipv6=1

# Second pair: veth30-veth33

sudo ip link add name veth30 type veth peer name veth33

sudo ip link set dev veth30 up

sudo ip link set dev veth33 up

sudo ip link set veth30 mtu 9500

sudo ip link set veth33 mtu 9500

sudo sysctl net.ipv6.conf.veth30.disable\_ipv6=1

sudo sysctl net.ipv6.conf.veth33.disable\_ipv6=1

=====================================================================================

p4c -b bmv2 pm\_switch.p4 -o pm\_switch.bmv2

sudo simple\_switch --interface 1@veth11 --interface 2@veth22 --interface 3@veth33 pm\_switch.bmv2/pm\_switch.json

=============================================================================

show\_tables

table\_info <TABLE\_NAME>

table\_add <TABLE\_NAME> <ACTION>

table\_dump <TABLE\_NAME>

=============================================================================

[cmd.txt]

table\_add forwarding pm\_forward 11.1.0.0/16 => 1

table\_add forwarding pm\_forward 22.2.0.0/16 => 2

table\_add forwarding pm\_forward 22.2.0.0/16 1 => 2

table\_add forwarding pm\_forward 11.1.0.0/16 2 => 1

mirroring\_add 250 3 // *mirroring\_add [mirror\_id] [output\_port]*

=============================================================================

sudo apt install -y python3-pip

pip3 install --upgrade setuptools

pip3 install --pre scapy[basic]

=============================================================================

sudo scapy

p = Ether()/IP(dst="22.2.0.1")/UDP() // TCP(dport=[80,443])

sendp(p, iface="veth11")

=====================================================================================

sudo tcpdump -n -i veth30

=====================================================================================

NOTE:

When a struct is inside of a header, the order of the fields for the purposes of extract and emit calls is the order of the fields as defined in the source code. An example of a header including a struct is included below.

**struct** ipv6\_addr {

**bit**<32> Addr0;

**bit**<32> Addr1;

**bit**<32> Addr2;

**bit**<32> Addr3;

}

**header** ipv6\_t {

**bit**<4> version;

**bit**<8> trafficClass;

**bit**<20> flowLabel;

**bit**<16> payloadLen;

**bit**<8> nextHdr;

**bit**<8> hopLimit;

ipv6\_addr src;

ipv6\_addr dst;

}